A Data Traffic Reduction Approach Towards Centralized Mining in the IoT Context

Ricardo Brandão, Ronaldo Goldschmidt and Ricardo Choren
Instituto Militar de Engenharia, Praça Gal Tibúrcio 80, Rio de Janeiro, Brazil

Keywords: Data Traffic Reduction, Data Summarization, Internet of Things, Distributed Data Mining.

Abstract: The use of Internet of Things (IoT) technology is growing each day. Its capacity to gather information about the behaviors of things, humans, and process is grabbing researchers' attention to the opportunity to use data mining technologies to automatically detect these behaviors. Traditionally, data mining technologies were designed to perform on single and centralized environments requiring a data transfer from IoT devices, which increases data traffic. This problem becomes even more critical in an IoT context, in which the sensors or devices generate a huge amount of data and, at the same time, have processing and storage limitations. To deal with this problem, some researchers emphasize the IoT data mining must be distributed. Nevertheless, this approach seems inappropriate once IoT devices have limited capacity in terms of processing and storage. In this paper, we aim to tackle the data traffic load problem by summarization. We propose a novel approach based on a grid-based data summarization that runs in the devices and sends the summarized data to a central node. The proposed solution was experimented using a real dataset and obtained an expressive reduction in the order of 99% without compromising the original dataset distribution's shape.

1 INTRODUCTION

In 2008, there already was the idea that the connected objects or devices could be active participants in the business processes since they would play a fundamental role in the interaction with consumers of services provided through the Internet (Haller et al., 2009). The term Internet of Things (IoT) refers to a set of technologies for accessing the data collected by various devices through wireless and wired Internet networks (Gubbi et al., 2013), and it expands the scope of the Internet, previously limited to computers (Miorandi et al., 2012). Currently, Iot is no longer a future strategic initiative for many organizations: it is a reality for many of them. 56% of organizations seeing it as strategic (IDC, 2016).

The evolution of the devices, becoming smaller, more precise and more integrated to the processes in which they participate, has made possible each day appear new applications for the IoT's devices. Thus these devices generate big data having useful, valuable and highly accurate data. However, it is difficult to extract the required information or data from the set of big data discovered by any device. Data management and analytics are critical to IoT-enabled solutions: for this purpose, data mining is used (Chovatiya et al., 2018).

Data mining techniques have traditionally been designed and developed to run in a centralized environment. For IoT use this means that it would be necessary to transfer huge amounts of data to a central system. Thus there would be an ever increasing data traffic on the network since 50 billion devices are forecasted to be connected to the internet. Data producers (devices) generate raw data and transfer it to the network to be mined in the central node. However, this structure is not sufficient for IoT (Shi et al., 2016). Data quantity at the edge is too large, which will lead to huge unnecessary bandwidth and computing resource usage. Also, most of the end nodes in IoT are energy constrained things, and the wireless communication module is usually very energy hungry, so offloading some computing tasks to the edge could be more energy efficient.

Given this problem, the present paper aims to propose an approach that reduces the traffic of data generated by IoT devices in order to enable the centralized mining of such data without compromising the shape of dataset's distribution. The idea is to use the concept of edge computing. Edge computing refers to the enabling technologies allowing computation to be performed at the edge of the network, putting some

Brandão, R., Goldschmidt, R. and Choren, R.
A Data Traffic Reduction Approach Towards Centralized Mining in the IoT Context.
DOI: 10.5220/0007674505630570
In *Proceedings of the 21st International Conference on Enterprise Information Systems (ICEIS 2019)*, pages 563-570
ISBN: 978-989-758-372-8

of the computing at the proximity of data sources (devices) (Shi et al., 2016). Since data transfer is energy hungry and data mining is heavy on processing (burdening the device), the solution relies on consolidating the data before it is transmitted. The solution divides the space in a grid so each cell of the grid is represented by the number of points contained therein and by its center of mass (Brandao and Goldschmidt, 2017). Each of the devices sends the representations of cells to a central node which, in his turn, consolidates all the information, allowing a summary representation of the complete dataset, since the grid's configuration is common to all nodes in the network.

The proposed approach was analyzed in a real smart city scenario. Indeed, IoT has become one of the most important types of infrastructure in smart cities since utilizing IoT technologies in a smart city can bring about a sustainable and pleasant living environment for its citizens (Park et al., 2018). However, a city populated by 1 million people will produce 180 PB data per day by 2019, contributed by public safety, health, utility, and transports, etc (Cisco, 2014). In the study scenario, data from 75,934 trips of 6 taxis were collected. The results show that it was possible to achieve a reduction in data traffic of around 99% without changing the overall format of the distribution in the complete and centralized dataset.

This work is organized in five more sections. Section 2 presets the basic concepts necessary to understand the proposed approach. Section 3 introduces the proposed approach to address the reduction of data traffic, as well as the structure of the prototype used for the execution of the experiments, whose results are presented and evaluated in section 4. Section 5 presents some works that deal with similar problems. Finally, Section 6 presents the conclusions and proposals for future work.

2 BACKGROUND

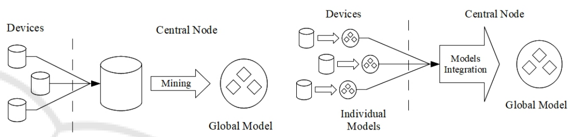
2.1 Big Data and Data Mining

The value of data is bound to the ability to extract knowledge from them. In this context, KDD (Knowledge Discovering in Databases) appears bringing techniques designed to address this problem. (Goldschmidt et al., 2015). With the ubiquitous integration of IoT devices with processes and people, associated with the Big Data tsunami, the KDD techniques have emerged as an important tool to discover and analysis of the behavior of processes and people monitored by these devices (Chen et al., 2015).

The KDD process involves the following steps: problem definition, which defines the data set to be analyzed, the application domain specialist and the application objectives; data pre-processing, which includes the functions of gathering, organization and data processing; data mining, the main stage of the KDD process, which the search for useful knowledge occurs; and post-processing, which covers the treatment of knowledge obtained in Data Mining, aiming to facilitate interpretation and evaluation by the specialist.

2.2 Mining of Distributed Data

In the IoT scenario, where data is distributed, there were two main approaches to use data mining: centralized and distributed as showed in the Figure 1.



(a) Centralized Mining (b) Distributed Mining

Figure 1: Data Mining.

In the centralized approach, each device gathers the raw data in a single environment, submit to the mining process generating a representative model of the mined data, as seen in Figure 1a. The main advantage of this approach is the use of mature data mining technologies that could deal with all data in the same location.

Distributed Mining is often used in situations where it is not convenient to move data across the network, like weakly coupled systems, systems with security and privacy issues (Goldschmidt et al., 2015). Each device gathers the data, runs the mining algorithms and generate individual models, sending them to the central node that processes the received models integrating then and generates a global model (Januzaj et al., 2004).

There is an important trade-off regarding mining the data generated by the IoT: either the centralized mining approach is chosen, which uses traditional and more mature techniques, but faces the challenge of increasing data traffic; or distributed mining that, despite the possibility of reducing traffic, is hampered by the limited resources of the devices, whether computational or storage, to perform mining operations.

A solution to be proposed to deal with this trade-off is the use of the centralized approach, running data reduction on the devices before sending to the centralized module. Vertical reduction techniques either require the active participation of the specialist or high computational power (Goldschmidt et al., 2015). The horizontal reduction techniques, besides have a lower computational cost, preserve the data schemas. In the context of IoT, which the devices have a low power of processing and storage, the horizontal reduction techniques rise as the most adequate.

3 PROPOSAL

With IoT, there would be a huge number of data generators in the network, but most of the devices would only periodically report sensed data to the network. Based on this observation, we understand that data could be preprocessed at the device level. Processed data will be sent to the network for future mining tasks execution. Thus, in the general structure of a KDD process depicted in Figure 2 with techniques used in the proposed approach are highlighted by solid lines. The solution acts at the pre-processing stage using summarization to reduce the amount of data and segmentation grouping sets of data that belongs to the same context. Normalization will be used to avoid comparing numbers with a wide variety of ranges.

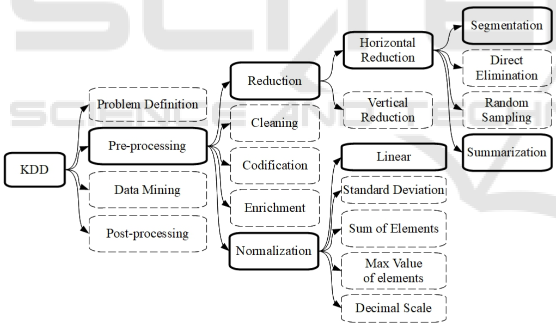


Figure 2: KDD Process.

The approach proposed in this paper presupposes the existence of q devices connected to a central node and running the same application. Where all the devices have a dataset with the same schema[1].

This approach has a multi-step process illustrated in Figure 3. Some of the processes occur in the central module and others in the devices. Additionally, the following observable variables represent the data gathered by each of the devices, where:

- $V = \{v_1, v_2, \dots, v_n\}$ is the complete dataset.
- d is an n-dimensional vector which the variable v_i represents each dimension.
- $p = 1, \dots, q$ are the devices used to gather data.

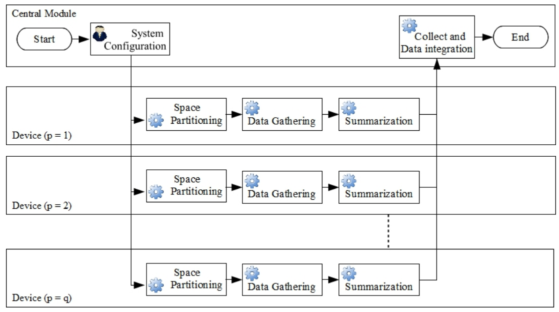


Figure 3: Complete Process.

The proposed process further considers that there is a domain specialist and/or a data analyst responsible for define segmentation rules and settings (Goldschmidt et al., 2015). The approach works as follows: in each device, the segmentation rules are applied and then summarize the data partitioning the space in a grid of cells, where each cell becomes represented by the number of points contained therein and by its center of mass (Brandao and Goldschmidt, 2017). In the sequence, the devices sent to the central node only the summarized information of cells with the number of points. Because the grid configuration is common to all nodes in the network, it is possible to summarize the complete dataset in the central node.

The following sections describe in detail the steps in this approach.

3.1 System Configuration

Not all variables will be used in the mining process, in addition, segments can be created in order to separate gathered data according to business rules defined by the specialist. This procedure aims to group variables and segments that are part of the same context. Thus, in this step, the first task is to define V_M and V_S, sets of variables to be used in mining and segmentation respectively, where:

$$V_M = \{v_{m_1}, v_{m_2}, \dots, v_{m_z}\} \mid V_M \neq \varnothing, V_M \subsetneq V$$
$$V_S = \{v_{s_1}, v_{s_2}, \dots, v_{s_w}\} \mid V_S \neq \varnothing, V_S \subsetneq V$$
$$V_M \cap V_S = \varnothing$$

Next, r segments are defined by the specialist, where each segment x is formed by a description and a filter as specified below:

$x.description$ = Text to describe content of segment x.
$x.filter$ = formula of predicates defined on V_S that defines the segment x dataset.

[1]The schema or structure of a dataset corresponds to the ordered set of attributes that the dataset contains. In the proposed approach, there is the assumption that this structure is common to the datasets of all devices in the IoT network.

Variables that will be mined can be expressed in a variety of units, ranges or scales. to avoid comparing numbers with a wide variety of ranges, they can be adjusted through normalization. Thus, for each variable of V_M the specialist must choose the maximum and minimum values to be used in the normalization process. The recommendation is that the values be chosen in such a way that the normalized values are in the range $0 < \alpha_i < 1$. However, this is not a restriction to the proposed model.

3.2 Space Partitioning

Is the first step performed on each device. It creates a grid with ε^z cells with edge size $= 1/\varepsilon$, where ε is a value defined by the specialist that divides each of the z dimensions into intervals of equal value.

Each cell γ in the grid is referenced in hyperspace by a z-dimensional coordinate system, as represented below.

$$\gamma(\omega_1, \omega_2, \ldots, \omega_z) \quad \text{where} \quad \omega_j \in \mathbb{Z}$$

ω_i are indexes that refer to cells. The Figure 4 illustrates an example with four cells in a bi-dimensional system.

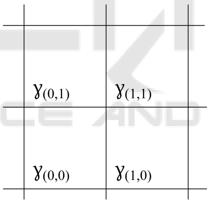


Figure 4: Cells representation in a bi-dimensional system.

The main goals of partitioning space into a grid of cells is to facilitate the identification of the position of a point within the space and reduce the number of comparisons of distance between points. This can have advantages in future mining processes, because it reduces its computational cost.

3.3 Data Gathering

If D is the dataset gathered by device p, for each segment x the corresponding filter is applied in order to select the records of D that represent the segment. On the result of the selection is applied the projection of the variables belonging to V_M, generating, thus, several tuples, each one represented by a vector $d_j(a_1, a_2, ..., a_z)$.

Depending on the characteristics of the dataset, the specialist can determine the need to perform a preprocessing, such as coding tasks, data cleaning, etc. The preprocessor routine generates new tuples, each one represented by the vector $dj(a'_1, a'_2, ..., a'_z)$, which in turn are submitted to the linear normalization that finally generates the segment instances $D(p,r) = \{\delta_1, \delta_2, ..., \delta_m\}$, where each $\delta_j(\alpha_1, \alpha_2, ..., \alpha_z)$ represents a normalized vector $dj(a'_1, a'2, ..., a'_z)$.

3.4 Summarization

This step is where the data volume reduction occurs. All points contained in a cell will be reduced to a single point located in the midpoint, or center of mass.

Each vector δ is contained in a γ cell. Thus, each cell γ will be represented by the number of z-dimensional vectors contained therein and a vector μ that corresponds to the center of mass of those vectors. An example with four two-dimensional cells is illustrated in Figure 5 which the filled dots represent the data and the empty points the centers of mass of the respective cells.

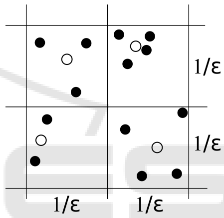


Figure 5: Grid with four cells. Filled dots are data, empty dots are the center of mass.

The equation 1 determines the coordinates of a cell γ which a vector δ must belong to.

$$\omega_i = int(\alpha_i \times \varepsilon) \text{where} 1 \leqslant i \leqslant z \tag{1}$$

The center of mass $\mu(\rho_1, \rho_2, ..., \rho_z)$ is calculated according the equation 2, where δ_i are the z-dimensional vectors belonging to γ and t the number of vectors of γ.

$$\mu = \frac{\sum_{i=1}^{t} \delta_i}{t} \tag{2}$$

Thus, each cell $\gamma(\omega_1, \omega_2, ..., \omega_z)$ is represented in a summarized way by the structure:

$\gamma.t$ = number of vectors belonging to the cellγ.
$\gamma.\mu$ = z-dimensional vector corresponding to the center of mass of γ.

The algorithm 1 executes the steps 3.3 and 3.4, run in the devices and has as input D_p, the dataset read in device p, the parameters for normalization and segmentation, and ε. As output the set $H(p,r)$ composed of cells γ belonging to segment x. For each record of D_p the algorithm checks which segment it belongs to. If a pre-processing step has been defined, it must be performed by the function *preProcessing* where the vector d is the input and the vector d' output. The vector δ, is the vector d' normalized. The function *calcCell* identifies the coordinates of the cell to which vector δ belongs. The last step of the algorithm is to include the vector in the cell through the function $\gamma_{(p,x)}.insertVector$ that increments the number of vectors in the variable $\gamma_{(p,x)}.t$ and calculates the new center of mass in the variable $\gamma_{(p,x)}.\mu$.

Algorithm 1: Summarization.

```
Input  : D_p - Collected data from p;
         Normalization parameters;
         ε
Output: {H_(p,x) | p is device and x is segment}
H_(p,x) ← ∅;
foreach element of D_p do
    x ← Segment to which the element belongs;
    d ← Projection of the variables belonging to V_C;
    d' ← preProcessing(d);
    δ ←
      normalization(d', normalization parameters);
    γ_(p,x) = calcCell(δ);
    if γ_(p,x) ∉ H_(p,x) then
        insert γ_(p,x) in H_(p,x);
    end
    γ_(p,x).insertVector(δ);
end
```

The complexity of the algorithm is $O(n)$ since it traverses the entire database without nested loops. It should be noted, however, that if the summarization routine can be inserted in the primary data gathering process, the complexity of the algorithm becomes $O(1)$.

3.5 Collect and Data Integration

In this step there is a superposition of the grids of all the devices. For each cell the new center of mass and the number of points will be calculated.

The central module receives from each device p, the set $H_{(p,x)}$, and create the set H_{C_x} which is composed of cells γ_{C_x}.

As all grids have the same configuration, a cell γ_{C_x} in the central node and a cell $\gamma_{(p,x)}$ in the device, that belongs on the same segment x will have the same coordinates. Then, when the central node receives the representation of $\gamma_{(p,x)}$, γ_{C_x} is updated according the equations 3 e 4.

$$\gamma_{C_x}.\mu = \frac{(\gamma_{C_x}.\mu \times \gamma_{C_x}.t) + (\gamma_{(p,x)}.\mu \times \gamma_{(p,x)}.t)}{\gamma_{C_x}.t + \gamma_{(p,x)}.t} \quad (3)$$

$$\gamma_{C_x}.t = \gamma_{C_x}.t + \gamma_{(d,x)}.t \quad (4)$$

Where $\gamma_{C_x}.\mu$ is the center of the mass of cell γ_{C_x} and $\gamma_{C_x}.t$ is the number of points of cell γ_{C_x}

4 EXPERIMENT

The experiment used a dataset with information gathered from taxis of Chicago, available on Google Big Query[2]. To facilitate the analysis and visualization, only two attributes were used to plot the result graphically. The attributes chosen were: trip duration in seconds and distance in miles.

In a metropolis like Chicago, weekend traffic is expected to be more fluid than on working days. Thus, two segments were created, one with data gathered on weekdays and the other with data from weekends.

So, in the configuration step were defined the following parameters:

V_S = {weekday}.
V_M = {Duration, Distance}
ε = 50

And segments x_0 e x_1:

$x_0.description$ = weekend trips
$x_0.filter$ = weekday = Saturday or Sunday

$x_1.description$ = working day trips
$x_1.filter$ = weekday between Monday and Friday

Six cabs where selected randomly. Each one has one device to gather data. The data characteristics are described in the Table 1.

Table 1: Sample Characteristics of dataset.

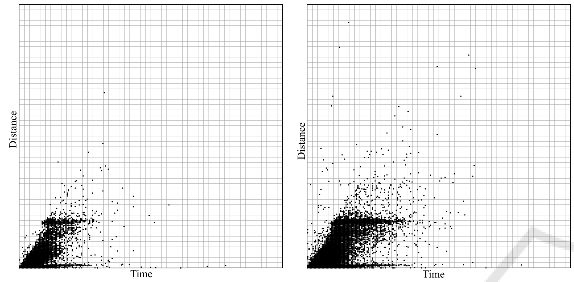
Car ID	Number of records	Size in Bytes
455b6b	14.209	978.090
4c8b67	11.246	764.151
5f1b23	13.688	940.907
7c51c6	10.006	690.567
b50eb9	14.097	971.251
d1b852	12.688	874.718
TOTAL	75.934	5.219.684

In the Figure 6, the original points are plotted gathered from all devices. In the proposed approach, only the cells containing points, with the information

[2]https://goo.gl/uPtj3y

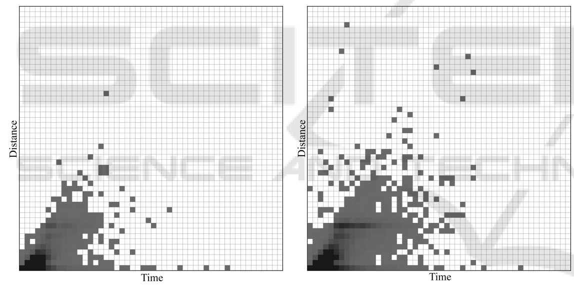
of the center of mass and the number of points, are sent to the central node. The Figure 7 represents the cells resulting from the process, which the cell shade represents the number of points: the darker, more points the cell has.

From the figures, it is possible to observe that, in the weekends, the trips have duration time shorter than the working days. The use of segmentation allowed the separation of different contexts for the same application, allowing a better analysis of the data generated.



(a) Weekends (b) Working Days

Figure 6: Chicago Taxis Dataset - Raw Data.



(a) Weekends (b) Working Days

Figure 7: Chicago Taxis - Summarized Data.

To evaluate the reduction rate on the data traffic (*reducRatio*) the volume of summarized data were compared with raw data, according the equation 5. Where $s_{rawData}$ is the size of raw data and $s_{sumData}$ is the size of summarized data.

$$reducRatio = 1 - \frac{s_{sumData}}{s_{rawData}} \times 100 \qquad (5)$$

To run the experiment, the raw data of each taxi was stored separately. The algorithm 1 was executed in each dataset and sent to another place that simulates the central node. To get the final result, a program gather all results provided by each device and integrated using equations 3 and 4.

The results are described in the Table 2. The "Raw Data" line indicates the total volume generated by the devices and the "Cell" line, the volume after running the algorithm 1. In this experiment, the algorithm runs twice, once for each segment.

Table 2: Data Reduction - Results.

Segment: Weekend	
Raw Data	1.376.312 bytes
Cells	15.623 bytes
Reduction Ratio	98,86%
Segment: Working days	
Raw Data	3.843.894 bytes
Cells	29.526 bytes
Reduction Ratio	99,23%
All Data	
Raw Data	**5.219.684 bytes**
Cells	**45.149 bytes**
Reduction Ratio	**99,14%**

The results obtained with the proposed approach led to a significant reduction of 99.14% of the data traffic that would be necessary if the data were all centralized. It is important to note that the reduction was obtained by preserving the shape of data distribution as can be seen by comparing the Figures 6 and 7.

5 RELATED WORKS

The work presented by Cantoni et al. (Cantoni et al., 2006) present the trade-off between the need to treat the data against the technical limitations of the sensors, especially with regard to energy consumption. Some solutions of distributed database and storage and data summarization are proposed as a strategy to reduce the size of transmitted data. They also make clear the important difference between decentralized and distributed algorithms. While in the decentralized algorithms each node is connected to all other nodes of the network, making the algorithm have a complexity $O(n^2)$, in the distributed algorithms the number of connections is equal to the number of nodes, so the complexity of the algorithm will be $O(n)$ or at most $O(n.log(n))$.

In another approach, with the objective of dealing with the large volume of data generated by the IoT devices, the work presented by Bin et al. (Bin et al., 2010) proposes four different models of data mining for the IoT: *(i)* multi-layer model, *(ii)* distributed data mining, *(iii)* data mining using grid computation, and (iv) an integration perspective of multi technologies.

The GDCluster algorithm (Mashayekhi et al., 2015) addresses the data traffic problem by clustering the data on each node and the propagation of its results through randomly chosen neighbor nodes, in a process called gossiping. Each node processes the internal data with the results received from their neighbors.

In another work (Bendechache and Kechadi, 2015), the authors also use clustering in order to reduce data traffic. The authors propose the D^2CA, an algorithm that uses clustering in a distributed way, maximizing parallelism and minimizing communication. In the proposed approach the distributed nodes execute clustering routines and create a representative model of their clusters based on their contours.

The distributed data clustering is also used as the basis of the work (Bendechache and Kechadi, 2015), where the authors propose the generation of local models and the integration of them into a global model in the central node. As a main difference, in this work the distributed nodes receive the global model, so that all network components have their data in a global context.

Using summarization as a technique for reducing data traffic, the paper (Brandao and Goldschmidt, 2017) adopts a centralized mining approach. It introduces the concept of space partitioning, creating cells or hypercubes with constant size edges. The summarization model also takes into account the processing and storage limitation of the devices.

The Table 3 displays a summary of related works, indicating whether is applied to the IoT context, centralized or distributed mining, and the approach used to reduce data traffic. Regarding the reduction of data traffic, the analyzed works use one of the following approaches: integration, approach where the process generates individual models that are sent to the central node where they are integrated; Clustering, which in this context indicates that mining is performed on the distributed modules and the result of the clustering is represented by a local model sent to the central module; and summarization which the data are reduced and represented in a concise manner.

Table 3: Related Works Summary.

Paper	IoT Application	Mining Approach	Data Traffic Reduction
(Cantoni et al., 2006)	Yes	Distributed	Integration
(Bin et al., 2010)	Yes	Distributed	n/a*
(Mashayekhi et al., 2015)	No	Distributed	Clustering
(Bendechache and Kechadi, 2015)	No	Distributed	Clustering
(Januzaj et al., 2004)	No	Distributed	Clustering
(Brandao and Goldschmidt, 2017)	Yes	Centralized	Summarization
This paper	Yes	Centralized	Summarization

* n/a - not applicable

It is possible to observe that several papers raise the problem of the massive generation of data by IoT devices and its impacts on the communication systems as one of the big challenges. In their proposals, they present as distributed mining solutions, in order to only send to the central node the data already treated and in many cases summarized through representation by local models.

In general, the approaches presented do not consider the fact that the poor processing and storage capacity of IoT devices can be a deterrent to the use of distributed clustering techniques, since the creation of local models requires resources that in not always viable in the heterogeneous devices. These works also disregard the fact that in most of their applications the devices have as main task the interaction with external environment. Transferring the responsibility of data mining to these devices may be impractical because it can not be interrupted the main tasks while the data mining process occupies computational resources.

6 CONCLUSION AND FUTURE WORKS

This article presented a proposal to reduce data traffic in the context of IoT, without burdening the processing of the devices. The proposed solution is based on partitioning the space into a grid of cells with a summarized representation of the points contained in each cell.

An experiment was performed using data generated by sensors installed in taxis that store travel information. The results obtained in the experiment showed a significant reduction of 99.14% considering the two segments used: weekend trips and working day trips. This is due to the repetitive behavior of the analyzed event, causing a single cell to have a large number of points, thus increasing the rate of reduction.

Analyzing the results it is also possible to verify that there is no significant loss in the dataset's shape when comparing with the original data (Figure 6), indicating that it is possible to rebuild the dataset on the central node, reducing data traffic without loss of meaning for future analysis. It's possible compare the proposed process as a reduction of image resolution, however, for each pixel of the new image, there is extra information: the number of original points and the center of mass.

So, it possible to observe that the use of the proposed approach reached its objective, obtaining an expressive reduction value in the data traffic with a process with complexity $O(n)$, that does not burden the processing of the devices.

Possibilities for future work include: researching data mining techniques in the central node, compare results obtained by data mining through the approach proposed in this article with traditional techniques, developing of behavior detection techniques from the data received in the central node, and developing models to detect noises and outliers.

REFERENCES

Bendechache, M. and Kechadi, M. T. (2015). Distributed clustering algorithm for spatial data mining. In *Spatial Data Mining and Geographical Knowledge Services (ICSDM), 2015 2nd IEEE International Conference on*, pages 60–65, Fuzhou, China. IEEE.

Bin, S., Yuan, L., and Xiaoyi, W. (2010). Research on data mining models for the internet of things. In *2010 International Conference on Image Analysis and Signal Processing*, pages 127–132, Zhejiang, China. IEEE.

Brandao, R. d. A. and Goldschmidt, R. R. (2017). Distributed data clustering in the context of the internet of things: A data traffic reduction approach. In *Proceedings of the 23rd Brazillian Symposium on Multimedia and the Web*, WebMedia '17, pages 313–316, New York, NY, USA. ACM.

Cantoni, V., Lombardi, L., and Lombardi, P. (2006). Challenges for data mining in distributed sensor networks. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 1, pages 1000–1007, Hong Kong, China. IEEE.

Chen, F., Deng, P., Wan, J., Zhang, D., Vasilakos, A. V., and Rong, X. (2015). Data mining for the internet of things: Literature review and challenges. *International Journal of Distributed Sensor Networks*, 11(8):431047.

Chovatiya, F., Prajapati, P., Vasa, J., and Patel, J. (2018). A research direction on data mining with iot. In Satapathy, S. C. and Joshi, A., editors, *Information and Communication Technology for Intelligent Systems (ICTIS 2017) - Volume 1*, pages 183–190, Cham. Springer International Publishing.

Cisco (2014). Cisco global cloud index: Forecast and methodology, 2014 – 2019.

Goldschmidt, R., Bezerra, E., and Passos, E. (2015). *Data Mining: Conceitos, técnicas, algoritmos, orientações e aplicações*. Elsevier Brazil, Rio de Janeiro, Brazil.

Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645 – 1660. Including Special sections: Cyber-enabled Distributed Computing for Ubiquitous Cloud and Network Services and Cloud Computing and Scientific Applications — Big Data, Scalable Analytics, and Beyond.

Haller, S., Karnouskos, S., and Schroth, C. (2009). The internet of things in an enterprise context. In Domingue, J., Fensel, D., and Traverso, P., editors, *Future Internet – FIS 2008*, pages 14–28, Berlin, Heidelberg. Springer Berlin Heidelberg.

IDC (2016). 2016 global iot decision maker survey.

Januzaj, E., Kriegel, H.-P., and Pfeifle, M. (2004). *DBDC: Density Based Distributed Clustering*, pages 88–105. Springer Berlin Heidelberg, Berlin, Heidelberg.

Mashayekhi, H., Habibi, J., Khalafbeigi, T., Voulgaris, S., and van Steen, M. (2015). Gdcluster: A general decentralized clustering algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 27(7):1892–1905.

Miorandi, D., Sicari, S., Pellegrini, F. D., and Chlamtac, I. (2012). Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7):1497 – 1516.

Park, E., del Pobil, A. P., and Kwon, S. J. (2018). The role of internet of things (iot) in smart cities: Technology roadmap-oriented approaches. *Sustainability*, 10(5).

Shi, W., Cao, J., Zhang, Q., Li, Y., and Xu, L. (2016). Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5):637–646.